Maastricht University

Fast ejection chain algorithms for vehicle routing with time windows

Citation for published version (APA):

Sontrop, H., van der Horn, P., & Uetz, M. J. (2005). *Fast ejection chain algorithms for vehicle routing with time windows*. METEOR, Maastricht University School of Business and Economics. METEOR Research Memorandum No. 013 https://doi.org/10.26481/umamet.2005013

Document status and date:
Published: 01/01/2005

DOI:
10.26481/umamet.2005013

Document Version:
Publisher's PDF, also known as Version of record

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy
If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Download date: 05 May. 2023

Fast Ejection Chain Algorithms for Vehicle Routing with Time Windows *

Herman Sontrop[1], Pieter van der Horn[1], and Marc Uetz[2]

[1] Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands. E-mail: {Herman.Sontrop,Pieter.van.der.Horn}@philips.com
[2] Maastricht University, Quantitative Economics, P.O. Box 616, 6200 MD Maastricht, The Netherlands. E-mail: M.Uetz@ke.unimaas.nl

Abstract. This paper introduces a new algorithm, based on the concept of ejection chains, to effectively target vehicle routing problems with time window constraints (VRPTW). Ejection chains create powerful compound moves within Local Search algorithms. Their potential to yield state of the art algorithms has been validated for the traveling salesman problem (TSP), for example. We show how ejection chains can be used to tackle the more general VRPTW as well. The yardstick behind ejection chain procedures is the underlying reference structure; it is used to coordinate the moves that are available for the Local Search algorithm via a given set of transition rules. Our main contribution is the introduction of a new reference structure, generalizing reference structures previously suggested for the TSP. The new reference structure, together with a set of simple transition rules, is tailored to handle the asymmetric aspects in a VRPTW. We use Tabu Search for the generation of the ejection chains, and on a higher algorithmic level, the ejection chain process is embedded into an Iterated Local Search algorithm. Computational results confirm that this approach leads to very fast algorithms, showing that ejection chain algorithms have the potential to compete with state of the art algorithms for the VRPTW.

1 Introduction

Recently, it has been shown that so-called *Stem & Cycle ejection chain* procedures can compete with state of the art implementations of the famous Lin-Kernighan algorithm [10] for solving large scale traveling salesman problems; see, e.g., [5]. This is remarkable, since Lin-Kernighan type algorithms had dominated this field for the last decades. Ejection chain procedures explicitly identify a so-called *reference structure*. This is a structure similar to, but slightly different from a solution, for example by violating certain types of constraints. Via a set of predefined *transition rules*, moves are generated from feasible solutions to reference structures, from one reference structure to another, and back from reference structures to solutions. This way the reference structure, together with the transition rules, define the moves that are available for a Local Search algorithm.

* This research was performed on behalf of the Centre for Quantitative Methods, CQM BV, P.O. Box 414, 5600 AK, Eindhoven, The Netherlands.

We address the vehicle routing problem with time window constraints (VRPTW). Given is a number of customers in the plane, with demands, a given service or delivery time, and a fleet of identical vehicles with known limited capacities. We are asked to find a set of routes starting and ending at a central depot, such that each client is served by exactly one vehicle. Clearly, any route must not violate the capacity constraints of the vehicle. In addition, each client must be serviced within its so-called *time window*. The time window specifies an earliest and a latest time at which the delivery must begin. If a vehicle arrives at a customer before the opening of the time window, the vehicle will have to wait. Arriving after the end of the time window is not allowed. Two different solutions with the same number of vehicles are usually ranked by the total distance travelled by the vehicles (sometimes, also the waiting time is taken into account). The objective considered in this paper is the total distance travelled by the vehicles, utilizing as many vehicles as required. We opted for this objective in order to be able to compare our results to known optimal solutions, which have been obtained using the same objective. In addition, we also experimented with a slightly modified approach where the usage of vehicles is penalized, in order to primarily drive down the number of vehicles[3].

The VRPTW has been extensively studied. The best *exact* procedures can still only handle small instances, small being in the order of 50–100 customers [16]. Meta-heuristic procedures mostly minimize the number of vehicles, and among solutions with the same number of vehicles, prefer those with small total distance travelled. Since the overall literature is extensive, we refer to [16] for a thorough introduction into vehicle routing in general, and many references. For concepts of Tabu Search, Simulated Annealing and other meta-heuristics, we refer to Aarts and Lenstra [1]. As a matter of fact, Tabu Search based procedures are the majority among the most effective algorithms for VRPTW, see, e.g., [14, 17, 9], but also other meta-heuristic frameworks proved to be effective; see [16].

The starting point of our research, motivated by practical interest, was the idea to generalize state of the art algorithms for the traveling salesman problem to the more general vehicle routing problem with time windows, in order to obtain good solutions very quickly. Previous research on the TSP has shown that Local Search algorithms –both Lin-Kernighan (LK) algorithms and Stem & Cycle (SC) ejection chain algorithms– are very effective when the available moves include the reversal of (sub)paths. However, this is generally not true for settings with time window constraints, because path reversals introduce time window violations. In this respect, it was already suggested by Glover in [7] that an augmentation of the SC reference structure, the *Doubly Rooted* (DR) reference structure, is more suitable for the asymmetric TSP.

Departing from this observation, our main contribution is a new reference structure for the VRPTW, generalizing Glovers DR reference structure for the TSP, that particularly targets the asymmetric nature of the VRPTW by avoiding path reversals. This reference structure lies at the heart of an ejection chain

[3] Clearly, these objective are correlated. When the capacities of the vehicles are large, however, solutions with less vehicles may lead to a larger total distance.

procedure that is based on Tabu Search. In addition to providing a new reference structure, a novelty in our contribution is also a study of different meta-heuristics that are used to steer the generation of ejection chains. (So far, ejection chain studies have mainly focused on the lower level of control, the ejection chain process itself.) It turned out that Iterated Local Search performed extremely well in terms of speed and solution quality. Our computational results on established standard test sets from Solomon [15] confirm that the resulting hybrid meta-heuristic is indeed fast and effective.

2 Ejection Chains and Reference Structures

Most heuristics directly move from one solution to another. A different approach is to first move to intermediate structures, *reference structures*, before moving to another solution. In such procedures a certain amount of infeasibility is introduced to the initial solution, which has to be 'ejected' in order to end up with a new feasible solution, usually called a *trial solution* in this context. The ejection of infeasibility can be delayed by moving to other reference structures first, creating a chain effect. At each level of such a chain, trial solutions are available by ejecting the infeasibility. Hence the term *ejection chain*. In the words of Glover [7], 'ejection chain procedures are based on the notion of generating compound sequences of moves, leading from one solution to another, by linked steps in which changes in selected elements cause other elements to be "ejected from" their current state, position or value assignment'. The yardstick behind such ejection chains is the underlying reference structure, which is a structure that resembles a solution, but is infeasible with respect to some of the constraints. The key idea is to introduce infeasibility by allowing certain vertices to have odd degrees, something that is not possible in a feasible VRP solution. The general structure of an ejection chain is depicted in Figure 1. In the example we see an

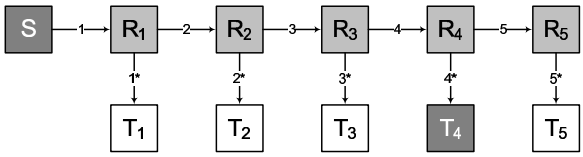


Fig. 1. Ejection chain example

ejection chain of five levels. Starting from a solution S we introduce a certain amount of infeasibility via move 1 resulting in reference structure R_1. From R_1 we can either eject the sustained infeasibility (move 1^*) or replace it by some other infeasibility (move 2). Obviously such a construction can be repeated, creating a chain. One can a priori set a maximum depth (in this case 5) and select the best trial solution seen in the chain. In this example the resulting compound move implicated by the chain consists of moves 1–2–3–4–4^*.

2.1 The Constrained Doubly Rooted reference structure

It can be considered both a strength and a weakness of state of the art TSP procedures that certain types of moves require a path reversal, i.e., the reversal of a sub path of the tour. In the absence of (tight) time windows, path reversals are an important building block of powerful moves. In the presence of time windows, however, path reversals often violate time windows, especially if the time windows are tight. Therefore, our idea to create an ejection chain procedure for the VRPTW is to have a reference structure that still provides a strong connectivity between solutions, but does not rely on path reversals in order to generate moves.

The reference structure we propose supersedes both the stem & cycle (SC) reference structure and the doubly rooted (DR) reference structure of Glover [7]. Figure 2 shows an instance of the SC structure. It is a spanning subgraph that consists of one cycle and a path. The path is called the *stem*, the end of the stem is called the *tip*. The vertex that is shared by the cycle and the stem is called the *root*. Note that the root and the tip have odd degrees, hence the reference structure does not represent a feasible solution to the problem. Figure 3 shows

Fig. 2. SC

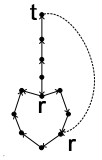


Fig. 3. DR

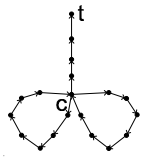


Fig. 4. Flower

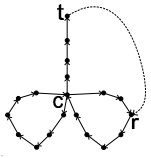


Fig. 5. CDR

a DR reference structure. It may be conceived as arising from an SC by adding an arc (t, j) which connects the tip to an arbitrary node j on the cycle. Again, this results in a structure that has two vertices that have odd degrees. In [7] Glover refers to both these vertices as *roots*. Glover shows that the DR structure has several advantages over the SC structure. He proves a connectivity result showing that the DR allows for more direct trajectories between solutions. Most important is that the connectivity result also holds for asymmetric problems. This is the main reason to base our new reference structure on the DR concept. In addition, the DR structure provides access to moves unavailable to an SC structure. Rego [13] generalizes the SC into the *Flower* reference structure[4], depicted in Figure 4. As can be seen it is an SC structure joined with multiple cycles. The intersection of the cycles is called the *core*. An SC structure can also be seen as a (trivial) Flower. The Flower concept proved to be very fruitful for VRP problems without time windows. The reference structure we propose is closely related to both the DR structure and the Flower concept. It may be conceived from a Flower by adding an arc (t, j) which connects the tip to an arbitrary node j on a cycle, and is shown in Figure 5. We call it a *Constrained*

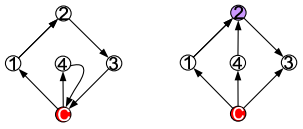
[4] Rego [13] refers to the vertex t as the *root*. We use the term *tip* as in [7], however.

Doubly Rooted reference structure (CDR). We use the word 'constrained' since the core, the depot vertex, always represents one of the roots. In Section 5 we briefly discuss an even more general, unconstrained version of the CDR. Note that a CDR, too, has two vertices that have odd degrees. We will refer to the vertex that represents the depot as the *core*, denoted by c. The other vertex with an odd degree is called the *root*, denoted by r. A vertex v such that the arc (v, r) exists is called a *subroot*. Note that when we eject such an arc (v, r) we obtain the Flower structure, therefore v could also be called an *implicit tip*.

2.2 Transition rules

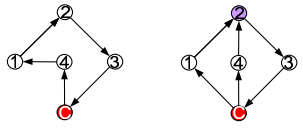
In order to exploit the reference structure, we need three types of *transition rules*. S-rules generate a CDR from a given start solution, E-rules generate one CDR from another, and T-rules generate a trial solution from a CDR. We state five simple transition rules that turn out to be sufficient.

S1-rule (Solution to CDR)
Eject (s, c)
Add (s, j) where $j \neq c$
and j not on path $c \rightarrow s$



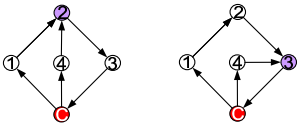
$(s = 4, j = 2)$

S2-rule (Solution to CDR)
Add (c, p) where $p \neq c$
Eject (q, p) where q is the predecessor of p
Add (q, j) where $j \neq c$ and $j \neq p$
and j not on path $c \rightarrow q$



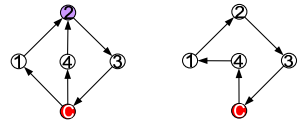
$(p = 1, q = 4, j = 2)$

E1-rule (CDR to CDR)
Eject (s, r) where $s \neq c$
Add (s, j) where $j \neq r$ and $j \neq c$
and j not on path $c \rightarrow s$



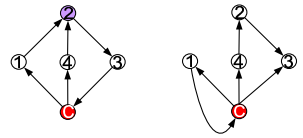
$(s = 4, r = 2, j = 3)$

T1-rule (CDR to Solution)
Eject (s, r) where $s \neq c$
Add (s, j) where $s \neq j$
Eject (c, j)



$(s = 4, r = 2, j = 1)$

T2-rule. (CDR to Solution)
Eject (s, r)
Add (s, c)



$(s = 1, r = 2)$

None of these rules involves a path reversal. The moves, however, can easily be adapted to include path reversals, if desired. In the figures, the core c is always the lowest vertex, and the root, if present, is displayed shaded. Returning to the example in Figure 1, move 1 would of type S, moves 2, 3, 4 and 5 of type E and moves 1^*, 2^*, 3^*, 4^* and 5^* would be of type T.

In principle, during the construction of an ejection chain, we always would like to compute the best possible move when using these rules. However, in order to speed up the procedure, whenever an arc is added in any of these rules, we do not check every possible arc, but restrict us to those arcs (v, w) that seem promising. To this end, we require that either vertex w must be a direct neighbor of vertex v in the *Delaunay triangulation* of the set of customers of the underlying instance[5], or (v, w) is one of the 12 shortest arcs leaving v such that this arc does not imply a time window violation on its own. (No conditions are placed on arcs that involve the depot vertex.) This proved to constitute a good candidate list for our test sets. Let us call a move, i.e., the application of a rule together with the selection of leaving and entering arcs, *admissible* if one of the above conditions for the entering arc is satisfied[6].

2.3 Ejection Chain Construction

The construction of an ejection chain now works as follows. Starting from a given solution, among all admissible S-moves, we select the one that leads to the best possible CDR. Then, we chain a sequence of E-moves from CDR to CDR in the same way, where ejected arcs are declared tabu for the remainder of the ejection chain. The chain is generated until either no more admissible E-moves exist that are not tabu, or a predefined maximum depth of the ejection chain is reached. From every single CDR, we generate all possible trial solutions using admissible T-moves. The *aspiration criterion* of the Tabu Search we use allows previously ejected arcs to be available in T-moves. Eventually, the compound move consists of moving to the best trial solution that has been generated.

In order to be able to move to the 'best' admissible CDR structure for any S- or E-transition, we must decide on the quality of a CDR reference structure (recall that it does not represent a feasible solution). To this end, we associate a simple cost function with the CDR reference structure, by just counting the total length of arcs included in the structure, incremented by a penalty term for the total time window violations of the *implicit routes*. For example, in Figure 5, we can distinguish three implicit routes: the two cycles as seen in Figure 4 and the route starting in c, via t and r back to c.

Finally, it is important to realize that the final algorithm generates several ejection chains after another. Each ejected arc is therefore not only declared tabu for the remainder of the given ejection chain, but all arcs ejected in one ejection

[5] See [6] for definitions regarding Delaunay triangulations.

[6] Glover uses slightly stricter conditions, so-called *legitimacy conditions*, to determine the arcs that are susceptible to being added or ejected, see [7]. The same legitimacy conditions have also been adopted by Rego [13].

chain are declared tabu also for the next θ ejection chains considered by the algorithm, where θ is a randomized parameter that is explained in more detail in Section 4. We observed that the procedure performed significantly better if all arcs ejected in the chain were declared tabu, instead of declaring tabu only the ejected arcs involved in the compound move. Apparently the ejected arcs in the remainder of a chain constitute so-called *critical event memory* of the Tabu Search process, see [8].

3 Higher Level Meta-heuristics

Ejection chain procedures are able to manipulate multiple solution components within a single compound move, as explained in the previous section. Based on the theory developed by Glover [7] for the DR reference structure, the CDR-structure proposed in this paper is conjectured to provide a strong form of connectivity between any two solutions. Therefore, a search can (and indeed did) easily get stuck in a so-called *basin of attraction*, see [11]. Hence we need a mechanism that will provide an escape trajectory from such a basin. We considered Simulated Annealing (SA) as well as Iterated Local Search (ILS) to steer the generation of ejection chains. SA requires that neighbor generation and feasibility checking is fast. We experimented with two SA designs, one that returned the best trial solution to an SA process and one that returned all trial solutions to an SA process. In our view, however, the generation of neighbors via ejection chains is too costly for SA to work well. Preliminary testing confirmed our conjecture. In great contrast, the use of ILS in combination with tabu driven ejection chains proved to be very successful. Therefore we restrict further discussions to our experiments using ILS.

3.1 Iterated Local Search

A simple, yet surprisingly effective technique is to apply an *Iterated Local Search* strategy, see [11, 2]. We propose an ILS where the embedded heuristic is based on tabu controlled ejection chains and uses random vertex ejections as *kick moves*. From a given start solution S_1 a basic ILS scheme can be stated as follows:

> Apply a kick move on S_1 yielding S_2.
> Perform a Local Search on S_2 resulting in S_3.
> Choose whether or not to accept S_3.
> If S_3 is accepted restart from S_3, otherwise restart from S_1.

We refer to the solution on which the kick move is performed as the *center solution*. We propose the random ejection of a single vertex v as a kick move. The ejected vertex v will form a new route on its own, while we link v's predecessor to its successor. Note that this kick move never introduces infeasibility. One of the main strengths of our procedure is its ability to merge routes in a way similar, but much more efficient, than the well known Clark & Wright savings procedure [3]. Therefore creating single vertex routes presents no problems. In fact, the kick move makes it much easier for our procedure to find new moves,

since ejecting vertices always creates more 'freedom' in existing routes. Note that, through the objective, the procedure is strongly biased to decrease the number of routes, so creating a new route by ejecting a vertex through a kick is a move not likely to be made by the procedure otherwise, even though the kick move is available as a forced combination of an $S2$- and $T2$-move. We thus follow the view expressed in [12], namely that a kick move should correspond to a modification of the structure that is not easily accessible to the moves already available or is unlikely to be chosen by the procedure itself. Since we use Tabu Search, instead of strict Local Search, the kick move is temporarily 'irreversible', thus the procedure is forced to find an alternative way to repair the sustained 'damage' obtained from the kick move. Algorithm 1 summarizes our proposed approach.

Algorithm 1: ILS Ejection Chain Procedure

```
 1 Generate Initial Solution
 2 while n < MaxNrOfIterations do
 3 |   while Depth < Maximum depth do
 4 |   |   Move to best available CDR reference structure
 5 |   |   Determine best available trial solution
 6 |   |   Depth++
 7 |   end
 8 |   Determine best trial solution over current ejection chain
 9 |   Update best known solution and center solution
10 |   n++
11 |   if n mod KickFrequency = 0 then
12 |   |   Return to center solution
13 |   |   Apply kick move
14 |   end
15 |   if No improvement found for α iterations then
16 |   |   Apply diversification move
17 |   end
18 end
```

Ejection chains are generated in lines 3–10. The maximum depth for an ejection chain was 50. Lines 11–14 describe the kick procedure; the kick move was performed every 5 iterations. Frequently performing a low strength kick worked best. Ejecting more than one vertex per kick move proved unsuccessful. Our ILS procedure only accepts improvements. Of course, other acceptance criteria could be used, see [4, 11]. The behavior of the ILS procedure is such that, in a sense, we always try to stay close to the current best solution value. This is in line with the so-called *Pyramid Principle*, as stated by Glover [8]. Finally, lines 15–17 provide a diversification measure that takes effect when the ILS itself is unable to improve solutions for a pre-specified number of iterations. To this end, recall that, after each ejection chain, all ejected arcs involved in S- and E-rules are declared tabu for an additional θ ejection chains, where θ is randomly drawn from the interval $[10, 30]$. The ejected arcs involved in the final T-move are also declared tabu. As a diversification measure, we increased this interval to [70,100] for 100 iterations, if the procedure did not find an improvement for 1000 subsequent iterations. No kick moves are performed during these 100 iterations. The diversification move ends by declaring the best solution in the 100th ejection chain as the new center solution.

4 Computational results

We present results for two versions of the algorithm, one that minimizes total distance only (DIST), irrespective of the number of vehicles used, and one that also minimizes total distance, but simultaneously penalizes the use of vehicles in the objective function (VEH). The second version was implemented to be able to compare with other heuristics, since they, in contrast to exact methods, primarily focus on minimization of vehicles.

Table 1 shows the computational results on the Solomon VRPTW test instances with 100 clients [15]. For some instances the global minimum distance is known [16], in which case we include the corresponding values in the column labelled 'Global Optima'. The table further shows, per instance, the ARO (discussed below), the results of the two variants of our algorithm, showing respectively the number of vehicles, total distance, and time (in seconds) when the best solution was found, as well as the solutions of another Tabu Search based algorithm (see below). Both versions of the algorithm share the same design, they only differ in the utilized objective function. We used trivial start solutions, in which each client forms an individual route. The algorithms were run for 10 minutes per instance, resulting in an average number of some 300,000 ejection chains per instance. The algorithm is coded in C++, using the Microsoft Visual C++ .NET compiler (2003), on a 2.8Ghz Pentium 4 processor, with 1GB RAM.

The results of Table 1 confirm that the algorithm gets close to the optimal solutions. For instances for which global optimal values are available, the results are on average 0.99% away from the global optimum. These values were found after 170 seconds on average (for the version that minimizes total distance, DIST). When we terminate the same algorithm after 60 seconds, the solutions are 1.67% away from the optimum, on average. Although the method is designed to minimize distance, we were able to obtain reasonable results for minimizing the number of vehicles, too, when we biased the objective function. We compare our results to those of another tabu implementation by Rochat and Taillard [14]. In a review of VRPTW designs in [16] from 2002, the results obtained by this method are referred to as excellent. It can be concluded that the algorithm labelled VEH compares reasonably to Rochat and Taillards results. It results in, on average, half a vehicle more, however with less total distance.

The column labelled ARO displays the average relative overlap (ARO) of any two time windows[7]. We observed that, on average, when the procedure performs worse in terms of solution quality and/or time required to find good solutions, the ARO is high. This makes sense, since a high ARO implies that the instance more closely resembles a VRP without time windows, while our procedure is explicitly designed for settings in which the time windows are tight. In fact, in settings with a lot of overlap between time windows, path reversals might be useful. But as a tribute to time windows, path reversals are explicitly excluded in our algorithm design.

[7] $\text{ARO} = 100 \cdot \sum_i \sum_{j>i} \left(\frac{Overlap_{ij}}{|TW_i|} + \frac{Overlap_{ij}}{|TW_j|} \right) / (n(n-1))$, where $|TW_i|$ is the length of time window i and n is the number of clients (the depot is not considered a client).

Solomon 100		EC MIN DIST			EC MIN VEH			R-T [14]		Global Optima	
SET	ARO	VEH	DIST	TIME	VEH	DIST	TIME	VEH	DIST	VEH	DIST
C101	6	10	828.9	1.2	10	828.9	0.3	10	828.9	10	827.3
C102	29	10	828.9	9.5	10	828.9	41	10	828.9	10	827.3
C103	53	10	828.1	28.7	10	828.1	312.5	10	828.1	10	826.3
C104	76	10	829.0	378	10	840.0	390.8	10	841.6	10	822.9
C105	12	10	828.9	6	10	828.9	10.2	10	828.9	10	827.3
C106	15	10	828.9	0.9	10	828.9	7.9	10	828.9	10	827.3
C107	18	10	828.9	62.8	10	828.9	18.3	10	828.9	10	827.3
C108	25	10	828.9	5.6	10	828.9	2.1	10	828.9	10	827.3
C109	38	10	828.9	58	10	828.9	10.7	10	828.9	10	827.3
C201	4	3	591.6	2.9	3	591.6	1.4	3	591.6	3	589.1
C202	28	3	591.6	19	3	591.6	4.5	3	591.6	3	589.1
C203	52	3	591.2	11.4	3	591.2	57.5	3	591.2	3	588.7
C204	76	3	590.6	308.2	3	594.5	594.4	3	597.8	-	-
C205	94	3	588.9	4.6	3	588.9	7	3	588.9	3	586.4
C206	15	3	588.5	8.1	3	588.5	3.1	3	588.5	3	586.0
C207	19	3	588.3	20.1	3	588.3	13.3	3	588.5	3	585.8
C208	20	3	588.3	25.6	3	588.3	7.2	3	588.5	3	585.8
R101	6	20	1642.9	34.4	19	1650.8	38	19	1656.2	20	1637.7
R102	28	18	1473.7	268.6	18	1473.7	420.7	18	1477.4	18	1466.6
R103	51	15	1227.8	320.6	14	1218.7	319.3	14	1222.9	14	1208.7
R104	74	11	1006.6	169.4	10	1007.8	432.2	10	1013.3	-	-
R105	19	15	1365.5	132.5	15	1367.5	114.2	14	1404.8	15	1355.3
R106	36	13	1246.1	514.9	13	1243.9	219.8	12	1293.9	13	1234.6
R107	56	12	1097.8	240.5	11	1092.0	24.8	11	1085.8	11	1064.6
R108	77	10	969.2	286	10	980.2	244.1	10	965.3	-	-
R109	38	13	1161.8	440.3	12	1172.6	253.6	12	1186.4	13	1146.9
R110	56	12	1108.2	493.3	12	1102.4	284.2	11	1107.9	12	1068.0
R111	55	12	1088.7	203.4	11	1092.2	47	11	1070.9	12	1048.7
R112	78	11	981.7	583.9	10	1000.2	314.5	10	965.7	-	-
R201	13	8	1150.0	390.7	6	1171.8	382.3	4	1485.4	8	1143.2
R202	34	7	1041.2	468.1	6	1049.5	599.4	4	1101.5	-	-
R203	55	6	877.6	182	5	890.0	437.7	4	913.0	-	-
R204	77	5	751.2	598.5	3	808.8	103.8	3	824.6	-	-
R205	28	6	966.7	216.6	4	1009.5	18.9	3	1205.6	-	-
R206	44	5	902.1	198.5	5	896.4	433.9	3	956.1	-	-
R207	62	4	813.1	387.4	3	856.0	472.7	3	814.8	-	-
R208	81	4	725.7	215.6	3	710.5	306.8	3	708.8	-	-
R209	41	5	870.5	572.8	5	881.4	346	4	901.9	-	-
R210	44	7	927.0	473.9	4	931.6	486.7	3	1087.3	-	-
R211	59	5	779.3	530.3	4	770.7	320.4	3	794.5	-	-
RC101	19	16	1645.6	20.2	15	1663.4	153.2	15	1737.0	15	1619.8
RC102	36	15	1487.6	532	14	1505.1	351.7	13	1480.7	14	1457.4
RC103	55	12	1289.1	206.1	13	1360.7	599.6	11	1264.3	11	1258.0
RC104	76	11	1187.0	142.2	11	1169.1	192.4	10	1157.2	-	-
RC105	36	16	1544.0	552.1	15	1562.7	274.7	15	1543.2	15	1513.7
RC106	40	14	1412.7	339.1	13	1404.4	428	12	1415.6	-	-
RC107	59	12	1235.2	580.8	11	1255.1	414.3	11	1262.4	-	-
RC108	75	12	1145.9	52.1	11	1162.9	259.6	11	1149.6	-	-
RC201	14	9	1268.8	285.7	5	1327.2	162.3	5	1469.7	9	1261.8
RC202	34	8	1113.6	197.1	5	1160.6	281.2	4	1443.7	-	-
RC203	56	6	955.4	146.3	4	1015.2	170.5	4	1014.0	-	-
RC204	77	4	791.4	207.2	4	812.1	438	3	843.1	-	-
RC205	28	7	1167.6	128	5	1290.0	144.5	5	1286.7	-	-
RC206	30	6	1080.1	418.1	5	1067.0	515	4	1207.8	-	-
RC207	44	7	988.2	74.5	5	996.5	589	4	1079.1	-	-
RC208	63	6	817.5	449	5	789.5	380.5	3	919.8	-	-
Averages		8.9	990.8	235.8	8.2	1002.0	240.3	7.7	1030.6	-	-

Table 1: Results on the Solomon VRPTW instances with 100 clients.

5 Conclusions and recommendations

One of the key reasons for the good performance of our algorithms for VRPTW is, we believe, due to the ability to generate powerful compound moves that do not require a path reversal. It must be noted, however, that methods that do perform path reversals, are extremely efficient in settings without time windows. We observed that, on average, when there is high overlap in the time windows, the procedure performs less effective. It is likely that the absence of path reversals is a reason. However, our concept can be adapted quite easily to include moves that use path reversals, too.

It can be considered a strength that the procedures do not, in any form, use pre-processing or post-processing. Also, the procedures cannot be considered two-phased methods, since they always use the same, extremely simple, start solution. A strong feature of the Iterated Local Search is that the kick move can be made very problem-specific, and can be used to decrease any possibly sustained infeasibility during the search.

Generalizing the CDR reference structure by relaxing the constraint that one of the roots must be the core is likely to further improve the procedure. The resulting *Generalized Doubly Rooted* reference structure (GDR) is shown in figure 6. The stated rules can be used as guidelines to create new S, E and T-type rules to exploit the GDR structure. E rules can be constructed that are able to increase or decrease the number of routes. In contrast, in the stated procedure, the total number of routes cannot be changed by more than one route per ejection chain. Further research is necessary to examine the potential of the GDR structure over the CDR structure. Finally, the procedure can, in

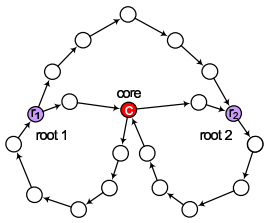


Fig. 6. Generalized Doubly Rooted reference structure (GDR)

all probability, easily be extended to the multi-depot case by using the reference structure in Figure 6, but allowing the roots to lie on cycles not necessarily joined by the same depot.

Acknowledgements. We thank Geert Teeuwen for the cooperation at CQM. In addition, the authors would like to thank Fred Glover and Emile Aarts for all their valuable comments, suggestions and encouragement. We found their collaboration very inspiring. Finally, we thank the anonymous referees for their valuable suggestions for improvement.

References

1. E. H. L. Aarts and J. K. Lenstra. *Local search in combinatorial optimization*. Wiley, Chichester, UK, 1996.
2. C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308, 2003.
3. G. Clark and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.
4. M. den Besten, T. Stützle, and M. Dorigo. Design of iterated local search algorithms. In E. J. W. Boers, J. Gottlieb, P. L. Lanzi, R. E. Smith, S. Cagnoni, E. Hart, G. R. Raidl, and H. Tijink, editors, *Applications of Evolutionary Computing*, volume 2037 of *Lecture Notes in Computer Science*, pages 441–451, 2001.
5. D. Gamboa, C. Rego, and F. Glover. Implementation analysis of efficient heuristic algorithms for the traveling salesman problem. *Computers and Operations Research*, 2005. To appear.
6. P. L. George and H. Borouchaki. *Delaunay Triangulation and Meshing, Applications to Finite Elements*. Hermes, 1998.
7. F. Glover. Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics*, 65:223–253, 1996.
8. F. Glover and M. Laguna. *Tabu Search*. Kluwer, Dordrecht, NL, 1998.
9. P. Kilby, P. Prosser, and P. Shaw. Guided local search for the vehicle routing problem. In S. Voss, S. Martello, I. H. Osman, and C. Roucairol, editors, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 473–486. Kluwer, Boston, MA, 1997.
10. S. Lin and B. W. Kernighan. An effective heuristic for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.
11. H. R. Lourenco, O. Martin, and T. Stützle. Iterated local search. In F. Glover and G. Kochenberger, editors, *The Handbook of Metaheuristics*, pages 321–353. Kluwer, Norwell, MA, 2002.
12. O. C. Martin and S. W. Otto. *Combining Simulated Annealing with Local Search Heuristics*, volume 63 of *Annals of Operations Research*, pages 57–75. 1996.
13. C. Rego. A subpath ejection method for the vehicle routing problem. *Management Science*, 44(10):1447–1459, 1998.
14. Y. Rochat and É. D. Taillard. Probalistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167, 1995.
15. M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35:254–265, 1987.
16. P. Toth and D. Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.
17. J. Xu and J. Kelly. A network-flow based tabu search heuristic for the vehicle routing problem. *Transportation Science*, 30:379–393, 1996.